

[カードベース暗号とその展開 (後編)]

～情報セキュリティ教育にも応用可能な身近な道具を利用した暗号技術～



⑥ 新たな道具を用いる物理的暗号

宮原大輝 電気通信大学



新たな道具の活用

カードベース暗号は、トランプカードのような物理的なカード組を用いて、秘密計算やゼロ知識証明といった暗号機能を実現する手法である。最も有名な例である「ファイブカードトリック」では、アリスとボブの2人が互いの気持ちを伏せたまま、デートに行くという合意（2入力論理積）が取れたか否かを決定できる。ここで重要なのは、アリスとボブは最初に自分の気持ちをカードの並びで表現し、裏向きの状態で場に置く点である。当然ながら、カードを表向きに置いてしまうと、即座に自身の気持ちが相手に漏えいし、これでは秘密計算となり得ない。裏向きのカードの絵柄は、物理的に外部からは見ることができないというカードの基本的な性質を基に、秘密計算が実現されている（余談だが「絵柄が異なるカードを裏返すと、まったく同じ見た目になる」という状態をデジタルの世界で再現するのは難しい。情報を隠すためには乱数を用いるのが一般的であるが、そうすると見た目（数値）はバラバラになり、逆に見た目を完全に一致させると、それは中身も同じであることを意味する）。

一方で本稿では、カード組以外の物理的な道具を用いる暗号プロトコル¹⁾を2つ紹介する。まず最初に、ボールと袋を用いる暗号プロトコルを紹介する。次に、重りと天秤を用いる暗号プロトコル、通称「天秤ベース暗号」²⁾を紹介する。なお、これら

は筆者を含む研究者グループによって開発された方式であるが、筆者が直接開発にかかわっていない方式の1つに、硬貨のようなコインを用いる「コインベース暗号」が存在する。詳細は文献³⁾を参照されたい。

ボールと袋を用いる暗号

前述したように「ファイブカードトリック」は5枚のカードを用いて2入力AND（論理積）の秘密計算を実現する。ここではまず、6つのボールと3つの袋を用いることで同様の秘密計算を実現できることを示す。

2入力ANDプロトコル

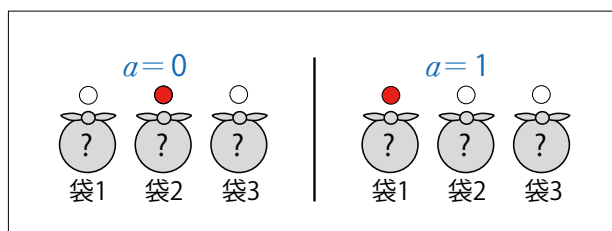
アリスとボブはそれぞれ、秘密のビット $a, b \in \{0,1\}$ に加えて、赤いボールを1つ、白いボールを2つ持つとしよう。ボールの大きさはすべて同一とする。3つの袋を便宜上、袋1,2,3と名付け、 $a=b=1$ （合意）の場合に限り、2つの赤いボールが同じ袋に入るように進む。詳細は次の通りである。

1. アリスはボブに見えないように、3つのボールを袋に入れる（図-1）。 $a=0$ の場合は袋2に赤いボールを入れ、ほかの2つの袋に白いボールを1つずつ入れる。 $a=1$ の場合は、袋1に赤いボールを入れ、ほかの袋に白いボールを1つずつ入れる。ただし袋の外側からは中のボールが見えないこと

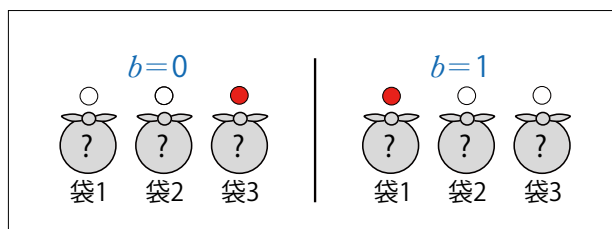
に注意しよう。

2. ボブはアリスと同様にボールを袋に入れる。ただし $b=0$ の場合は袋3に赤いボールを入れ、 $b=1$ の場合は（アリスと同じく）袋1に赤いボールを入れる（図-2）。ただしボブはこのとき、中に入っているボールの色を見てはいけない。
3. ここで3つの袋には $a=b=1$ ，すなわち $a \wedge b=1$ の場合に限り、2つの赤いボールが同じ袋（袋1）に入っていることに注意しよう。そこで3つの袋をランダムに並び替え、シャッフルする。
4. 袋からボールをすべて取り出す。2つの赤いボールが同じ袋から出てきた場合は $a \wedge b=1$ ，そうでない場合は $a \wedge b=0$ である。

このプロトコルがなぜ秘密計算となるのかを示す。手順2においてボブが袋にボールを入れた際、 a, b の値に応じて次のように4パターンが存在する（図-3）。 $a \wedge b=1$ （図-3の右下）の場合のみ赤いボールが同じ袋に入っているのは、手順3で確認した。 $a \wedge b=0$ の場合である3パターンについては、どれもボールと袋の内訳が同じであることが分かる。すなわち2つの袋に赤白が含まれ、もう1つの袋には白白が含まれている。したがって3つの袋をシャッフルして（手順3）ボールを取り出すことにより（手順4）、 $a \wedge b$ の値のみが判明する。



■図-1 アリスによるボールの入れ方



■図-2 ボブによるボールの入れ方

特に $a \wedge b=0$ の場合は、シャッフル後のボールと袋の内訳から a, b の値を推測することはできず、これにより秘密計算が成り立つ。

本稿では以上のようにボールと袋を用いて説明したが、同文献²⁾では具材（人参と玉ねぎ）と鍋を用いた「料理する暗号学者の問題（Cooking Cryptographers Problem）」として動機付けがなされている。これは著名な「食事する暗号学者の問題（Dining Cryptographers Problem）」をオマージュしたものである。

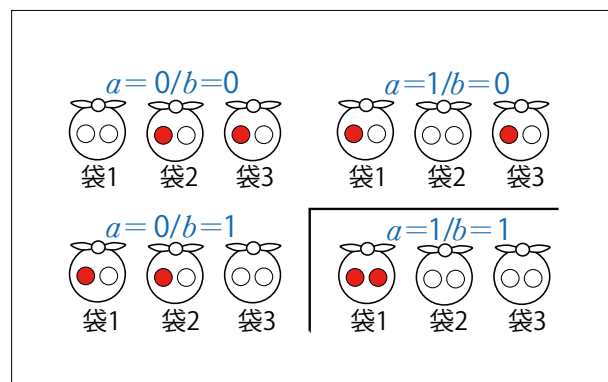
モチベーション

このボールと袋を用いる2入力ANDプロトコルに対して、理論的な観点から以下の課題が提起される。

1. AND以外の任意の論理関数への拡張。
2. 実行に要するボールおよび袋の数の最小化。
3. プロトコルを厳密に記述して正当性や安全性を検証するための計算モデルの構築。

本稿ではこれらの課題のうち、1つ目の問いへの取り組みを次節で紹介する。なお、同文献²⁾では3つ目の問いである計算モデルの構築も検討されているが、本稿では紙幅の都合上その詳細は割愛する。また、2つ目の問いに関しては同文献でも言及されておらず、特に前述の2入力ANDプロトコルで用いた「6つのボールと3つの袋」という構成が最小数であるか否かは、いまだ明らかになっていない。

ここでカードベース暗号との相違点について述べ



■図-3 2入力ANDプロトコルにおける内訳

る。まず重要な点として、「袋に色付きのボールを1つ入れた状態」は、物理的に「同じ色のカードを裏返して置いた状態」と同一である。したがって、有名な「ファイブカードトリック」などのカードベースプロトコルは、一見するとボールと袋を用いても実装可能に思える。しかしカードベースプロトコルにおいて根幹をなす「ランダムカット」のような特殊なシャッフルを、袋に対しても再現するには相応の工夫を要する(ランダムカットについては前号に掲載された水木敬明氏による特集記事を参照されたい)。すなわち、カードベース暗号をボールと袋で再現するのは容易でない。一方で、「ボールと袋を用いる暗号」をカード組で再現することはきわめて容易である。たとえば前述した2入力ANDプロトコルは、アリスとボブがそれぞれ赤のカードを1枚、白のカードを2枚持ち、ボブはアリスが置いたカードに自身のカードを重ねて「カードの組」を作るだけで、袋の中にボールが入った状態を完全に再現できる。ただしこれは計6枚のカードを必要とするため、5枚のカードで計算する「ファイブカードトリック」に比べて効率性は劣る。以上の議論より、袋とボールを用いる暗号には独自の理論的意義が認められ、研究すべきテーマであると言える。

多入力 AND プロトコル

$n(>2)$ 人のプレイヤーが存在し、 i 番目のプレイヤー P_i は秘密のビット $x_i \in \{0,1\}$ を持つとする。赤いボールを2個、白いボールを $2n$ 個、袋を $n+1$ 個準備し、 i 番目の袋を袋 i と名付ける。 n 入力 AND プロトコルは、以下のように動作する。

1. P_1 および P_2 は、前述の2入力ANDプロトコルと同じようにボールを袋に入れる。すなわち袋1,2,3に合計で赤いボールが2個、白いボールが4個入る。
2. $i=3$ から n まで以下の動作を繰り返す。
 - (a) 袋1から袋 i までをシャッフルする。シャッフルによってランダムに並び替えられた袋に

対して、 j 番目の袋を袋 j と名付ける。

- (b) 袋1からボールを1つ取り出し、その色が赤であれば、 P_i はその赤いボールに加えて白いボールを2個持つ。取り出したボールの色が白である場合は、そのボールを袋1に戻し、手順2(a)に戻る。
 - (c) P_i はほかのプレイヤーに見えないように、3つのボールを袋に入れる。 $x_i=0$ の場合は赤いボールを袋 $i+1$ に入れ、白いボールを袋1に入れる。 $x_i=1$ の場合は、赤いボールを袋1に入れ、白いボールを袋 $i+1$ に入れる。もう1つの白いボールは袋 $i+1$ に入れる。
3. ここで $n+1$ 個の袋には、 $x_1 \wedge x_2 \wedge \cdots \wedge x_n=1$ の場合に限り、2つの赤いボールが同じ袋(袋1)に入っていることに注意しよう。すなわち、2入力ANDの際と同じ状況となっている。そこで $n+1$ 個の袋をシャッフルする。
 4. 袋からボールをすべて取り出し、赤いボールが同じ袋から2個出てきた場合は $x_1 \wedge x_2 \wedge \cdots \wedge x_n=1$ 、そうでない場合は $x_1 \wedge x_2 \wedge \cdots \wedge x_n=0$ である。

このプロトコルの工夫は、手順2での P_3 以降の入力にある。手順2(b)によって、 $x_1 \wedge \cdots \wedge x_{i-1}=1$ の場合に限り、袋1に入っている残り1つのボールの色は赤となる。したがって、手順2(c)において $x_i=1$ であれば、袋1に赤いボールが入ることになり、2個の赤いボールが同じ袋に存在するという状態が維持される。一方で誰か一人でも0のビットを持っていれば、赤いボールが別の袋に入ってしまう、同じ袋に入ることはなくなる。各袋にボールが2つ入り、赤いボールが同じ袋に存在しないとすると、各袋の内訳は、赤白を含む袋が2個、白白を含む袋が $n-1$ 個となる。したがって $x_1 \wedge x_2 \wedge \cdots \wedge x_n=0$ の場合に(シャッフル後の)各袋のボールの内訳から入力値を推測することはできず、秘密計算が成り立つ。

このプロトコルは手順2(b)で、確率的に動作が

特集 Special Feature

進む。取り出したボールの色が赤である確率は、 $\frac{1}{7}$ である。したがって手順 2(b) を繰り返す回数の期待値は i 回であり、全体としては

$$\sum_{i=3}^n i = \frac{1}{2}n(n+1) - 3$$

となる。この回数の削減や、確率的にではなく決定的に動作するプロトコルの開発が課題である。また、用いるボールと袋の数の削減も興味深い問題である。

天秤ベース暗号

本章では、重りと天秤を用いる天秤ベース暗号を紹介する。これは有名な「偽コイン問題」に着想を得たものである。偽コイン問題には多くのバリエーションがあるが、ここではシンプルな設定を扱う。すなわち、外見がまったく同じ 8 枚のコインの中に 1 枚だけ軽い偽コインが混ざっている状況で、天秤を用いてその偽コインを特定するのが目的である (図-4)。

まず、天秤を 3 回使えば偽コインを見つけられるのは明らかである。8 枚を半分ずつ天秤に乗せ、軽かった方の 4 枚をさらに半分に分ける作業を繰り返せばよい。しかし、この問題の興味深い点は、天秤をわずか 2 回用いるだけで偽コインを特定することにある。その鍵は、4 枚ずつではなく 3 枚ずつ天秤に乗せるという工夫にある。1 回目の比較において、天秤の傾きに依拠して次のように場合分けを行う。

(a) 天秤が釣り合った場合、天秤に乗せた 6 枚には偽コインは含まれず、残りの 2 枚の中に偽コ

インが存在するはずである。したがって、この 2 枚を 2 回目に比較することで、偽コインを特定できる。

(b) 天秤が傾いた場合、軽い方の 3 枚の中に偽コインが含まれるはずである。この 3 枚のうち 2 枚を 2 回目に比較することで、それらが釣り合えば残りの 1 枚が、傾けば軽い方の 1 枚が偽コインであると特定される。

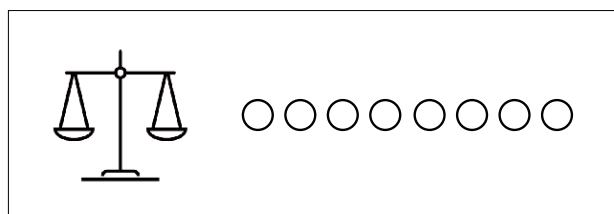
以上のように偽コイン問題では、天秤が左右に傾く状態に加えて、釣り合うという 3 つの状態を活用するのがポイントである。一方で、本稿のテーマである暗号学的な視点に立つと、偽コイン問題では「偽コインと本物のコインの外見上の区別がつかない」という物理的な性質が大前提となっている。天秤ベース暗号の着想は、この性質を逆手に取り、秘密情報をコインの重さとして表現することにある。すなわち、天秤による比較操作を演算と見なし、外見からは情報を秘匿したまま、何らかの暗号学的な計算を行えないか、という試みである。

これまでの成果

この天秤ベース暗号に関して、これまでに 4 件の国際会議論文が発表されている。このうち 3 件は筆者を含む研究グループによる成果である。

1. 数独といったペンシルパズルの解に対するゼロ知識証明 (1 件)。
2. 選挙のような投票に対して、投票内容を秘匿したまま結果のみを算出する投票プロトコル (1 件)。
3. AND 関数、多数決関数および任意の関数を計算するプロトコル (2 件)。

本稿ではこのうち、1 つ目のゼロ知識証明を紹介する。なお、これら以外にも、暗号に関する国内シンポジウムである SCIS 2026 において、別のペンシルパズルへの応用や、物理的なコインと天秤を用いた実装上の課題に関する報告が行われるなど、本分野のさらなる発展が期待される。特に単純な比較操作のみによるさまざまな暗号機能の実現は興味深い。



■図-4 偽コイン問題の設定

ゼロ知識証明

数独（SUDOKU）は（株）ニコリの登録商標であり、現在では世界で最も普及しているペンシルパズルといっても過言ではない。同社の Web ページで紹介されている例題（図-5 左）をもとに、ルールを確認しよう。数独は 9×9 の盤面で行われ、いくつかのセルにはあらかじめ 1 から 9 の数字が配置されている。太線で囲まれた 3×3 の領域をブロックと呼ぶ。数独の目的は、各行・各列・各ブロックにおいて、1 から 9 までの数字がちょうど 1 つずつ現れるように、空きマスに数字を埋めることである。答え（図-5 右）を見ると、たとえば 5 行目にはすでに 8 種類の数字が現れているため、残る中央の空きマスには、まだ現れていない 5 を埋めることになる。

ゼロ知識証明は簡潔に述べると、ある問題の解を秘密にしたまま、その問題に対して正当な解が存在することを、証明者が検証者に納得させる暗号プロトコルである。これを数独に当てはめると、各行・各列・各ブロックに数字が 1 つずつ現れる正しい割り当て（解）が存在することを、具体的な数字の配置を一切明かさずに証明することを意味する。これを物理的なコインと天秤を用いて実現する「天秤ベースゼロ知識証明プロトコル」の手順を解説する。

配置フェーズ

まず準備として、9 種類の異なる重さを持つコイ

ンを、種類ごとに 9×3 枚ずつ、計 243 枚用意する。各コインの重さは、1 から 9 までの各数字に対応している。解を知る証明者は、自身の持つ解に従って、 9×9 の盤面の各セルに対応する重さのコインを 3 枚置く。ここで前述した物理的な性質により、外部からは各コインの重さの判別がつかず、検証者に解が漏れることはない。

検証フェーズ

盤面に配置された 243 枚のコインに対し、証明者は以下の操作を通じて、それらが数独のルールを満たしていることを証明する。

1. 各セルに置かれた 3 枚のコインが、すべて同じ重さ（＝同じ数字）であることを示す。具体的には、各セルにおいて「1 枚目と 2 枚目」および「2 枚目と 3 枚目」を天秤で比較し、いずれも釣り合うことを示す（余談であるが、2007 年に提案された最初の「数独に対するカードベースゼロ知識証明プロトコル」では、同様に各セルへ 3 枚のカードを配置する手法がとられていた。しかし、カードベースでは秘密を保ったまま「3 枚が表す数字が同一であること」を証明することができず、結果として健全性エラーが生じる原因となっていた。健全性エラーについては前号に掲載された品川和雅氏による特集記事を参照されたい）。
2. 1 行目に置かれた 9 枚のコインが、すべて異なる重さである（すなわち 1 から 9 が重複なく現

8					5	1		
		1				8		
	4		2				9	
								2
1	2	3	4		6	7	8	9
6				1				
	8				9		5	
		2				4		
		7	6					1

8	3	9	7	6	5	1	2	4
2	6	1	3	9	4	8	7	5
7	4	5	2	8	1	3	9	6
5	9	4	8	3	7	6	1	2
1	2	3	4	5	6	7	8	9
6	7	8	9	1	2	5	4	3
3	8	6	1	4	9	2	5	7
9	1	2	5	7	3	4	6	8
4	5	7	6	2	8	9	3	1

■図-5
数独の例題（左）とその答え（右）

特集 Special Feature

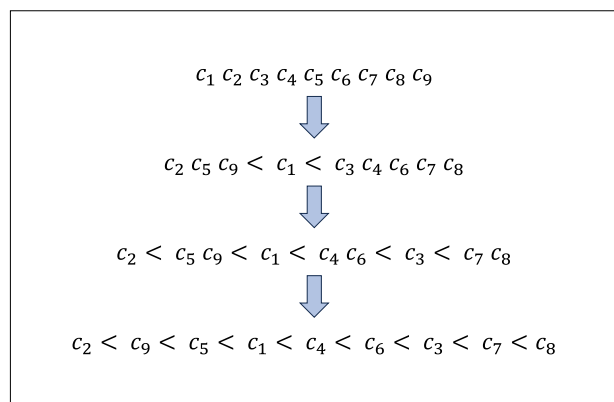
れる)ことを証明する。これには、クイックソートのようなソーティングアルゴリズムを応用する。具体的には次の通りである。

- (a) 9枚のコインをかき混ぜることにより、シャッフルする。
- (b) 天秤を比較演算器として用いてクイックソートを実行する。この際、一度も「釣り合う(=同じ重さである)」ことがないことを確認し、最終的に9枚のコインが重い順(あるいは軽い順)に一列に並ぶことを示す。図-6以下に各コインを c_i としたソーティングの具体例を示す。まず c_1 をピボットとして残りのコインと1枚ずつ比較し、 c_1 より「軽いグループ(3枚)」と「重いグループ(5枚)」の2つに分かれたとする。この時点で、 c_1 は全体で4番目の重さであることが確定する。分割された各グループにおいても、同様にピボットを1枚定めて残りのコインと比較し、さらに小さなグループへと分けていく動作を再帰的に繰り返す。

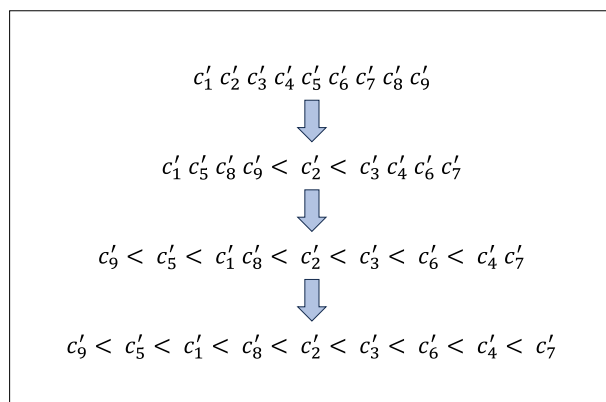
3. 残りの各行・各列・各ブロックに置かれた9枚のコインに対しても、同様にクイックソートを実行する。ただしここで、重要な注意点がある。単に「9枚のコインがすべて異なる重さである」ことを確認するだけでは不十分である。なぜなら、悪意のある証明者が配置フェーズにおいて、本来

存在しない10種類目(あるいはそれ以上)の重さを持つ不正なコインを紛れ込ませている可能性があるからである。すなわち、数独のルールを満たすためには、9枚のコインの重さに重複がないだけでなく、「1から9の数字の集合と一致している」ことを保証しなければならない。そのために、前手順でソートされた1行目のコイン c_i を活用し、以下の手順を踏む。

- (a) 検証対象となる9枚のコインをシャッフルする。シャッフル後の各コインを c'_i と表記する。
- (b) 各 c'_i に対してクイックソートを実行する際に、ピボットとして1行目のソート済みの各コイン c_j を指定するのが工夫である。具体例を以下図-7に示す。まずピボットとして、1行目の9枚のコインの中で中央の重さを持つコイン(手順2(b)の例では c_4)を指定し、各 c'_i と比較する。このとき各 c'_i が数独のルールを満たしているのであれば、それらは以下の3つのグループに過不足なく分割される。
 - ピボットより「軽いグループ」: 4枚
 - ピボットより「重いグループ」: 4枚
 - ピボットと「同じ重さのコイン」: 1枚
 このように、比較のたびに「ピボットと等しい重さが1枚だけ存在する」ことを確認しながら、再帰的に「軽い4枚」をさらに1行目の2番目の重さのコインで分割していく。



■図-6 クイックソートの具体例



■図-7 クイックソートにおける工夫の具体例

特集 Special Feature

安全性および効率性

このゼロ知識証明プロトコルが解に関する情報を一切漏らさないこと（ゼロ知識性）を確認する。各行・各列・各ブロックの検証において、天秤による比較を行う直前に9枚のコインがシャッフルされる。この操作により、「特定のセルのコイン」と「特定の重さ（数字）」の対応関係が失われる。結果として検証者が得られる情報は「その9枚の中に1から9に対応する重さが重複なく存在すること」のみに限定される。これは数独のルールそのものであり、ゼロ知識性が満たされる。

このプロトコルは、盤面サイズを $n \times n$ に一般化した数独に対しても容易に拡張できる。この際に必要となるコインの数は $3n^2$ 枚である。天秤の使用回数は $O(n^2 \log n)$ のオーダーで抑えられ、これは $O(n \log n)$ であるクイックソートを $3n$ 回繰り返すためである。

さらなる効率化

このゼロ知識証明プロトコルのさらなる効率化として、各セルに配置されるコインをわずか1枚で済ませる手法が SCIS 2026 で考案されている。紙幅の都合上、本誌ではその概略を紹介し、結びとする。

1. 各列・各ブロックに配置された9枚のコインに

ついて、9枚の総重量がそれぞれ等しいことを天秤で示す。この操作は「総重量のみ」の比較であるため、個別の重さを特定することなく、比較後に9枚を元のセルに戻すことができる。

2. 1行目に対して、前述のプロトコルと同様に、シャッフルとクイックソートを実行する。
3. 2行目から9行目の各行に対しても、同様にシャッフルを行った上で、手順2でソート済みの1行目のコインをピボットとしてクイックソートを実行する。

参考文献

- 1) Miyahara, D., Komano, Y., Mizuki, T. and Sone, H. : Cooking Cryptographers: Secure Multiparty Computation Based on Balls and Bags, IEEE Computer Security Foundations Symposium, pp.389-404 (2021).
- 2) Kaneko, S., Lafourcade, P., Mallordy, L., Miyahara, D., Puys, M. and Sakiyama, K. : Balance-Based ZKP Protocols for Pencil-and-Paper Puzzles, Information Security, Vol.15257 of LNCS, pp.211-231 (2025).
- 3) Komano, Y. and Mizuki, T. : Coin-Based Secure Computations, Int. J. Inf. Secur., 21, pp.833-846 (2022).

(2026年3月11日受付)

■宮原大輝（正会員） miyahara@uec.ac.jp

電気通信大学助教。専門は理論計算機科学、情報セキュリティおよびカードベース暗号。2021年に東北大学大学院情報科学研究科で博士（情報科学）を取得。東北大学ポスドク（学振PD）を経て同年より現職。

